

Cours 3

- Tableaux
- Tableaux à 2 dimensions
- Tableaux d'objets
- Classes String et StringBuffer

Les **tableaux** ne sont pas ni des objets ni des types simples.

Un tableau se rapproche d'un objet

- Il est manipulé par référence (adresse)
- Il nécessite de *new* pour être défini

Déclaration

`char tableau[]` ou `char[] tableau`

Création

- En utilisant *new*
`int tab[]; tab=new int[5];`
`char t2[]= new char[4];`
- implicitement avec l'initialisation
`int t2[]={1,2,3};`

- La taille est fixée
- On peut accéder à la taille par le champ **length**
- Les indices entre 0 et length-1
- Dépassement d'indice : lancement d'exception *ArrayIndexOutOfBoundsException*
- Un objet ou un tableau qui n'est plus référencé est récupéré par ramasse-miettes (garbage)

```
public class TabAffec
{public static void main(String args[])
    {char t1[]={ 'b', 'o', 'n', 'j', 'o', 'u', 'r' };
    char t2[]={ 'h', 'e', 'l', 'l', 'o' };
    char t3[]={ 'x', 'x', 'x', 'x' };
    t3=t1; t1=t2; t2=t3;
    System.out.print("t1 =");
    for (int i=0;i<t1.length;i++) System.out.print (t1[i]);
    System.out.println();
    System.out.print("t2 =");
    for (int i=0;i<t2.length;i++) System.out.print (t2[i]);
    System.out.println();
    System.out.print("t3 =");
    for (int i=0;i<t3.length;i++) System.out.print (t3[i]);
    System.out.println();
    }
}

t1=hello
t2=bonjour
t3=bonjour
```

Les paramètres du programme

```
class Prog23 {  
    public static void main (String args[])  
    {int i;  
     System.out.println("les arguments :");  
     for (i=0; i<args.length ; ++i)  
         System.out.println(args[i]);}  
}
```

Exécution

```
java Prog23 "Bonjour coucou " a b
```

les arguments :

Bonjour coucou

a

b

POO

5

```
class UtilTab  
{static double somme(double[] t)  
    {double s=0;      int i;  
      for (i=0;i <t.length; i++) s+=t[i];      return s;}  
  
static void affiche (double[] t)  
{for (int i=0; i <t.length; i++) System.out.print(t[i]+" ");  
 System.out.println();}  
  
public static double[] somme (double[] t1, double[] t2)  
{  
    int n=t1.length;  
    if (n!=t2.length) return null;  
    double [] res= new double[n];  
    for (int i=0; i <n; i++) res[i]=t1[i]+t2[i];      return res;}  
}  
public class TstUtil  
{ public static void main(String[] args)  
    { double tab1[]={1.5, 2.25, 3.5, 4.0};  
      System.out.println("tab1 "); UtilTab.affiche(tab1);  
      System.out.println("somme" + UtilTab.somme(tab1));  
      double tab2[]={2.5, 1.5, 3.5, 4.5};  
      System.out.println("tab2 "); UtilTab.affiche(tab2);  
      double [] tab3= UtilTab.somme(tab1,tab2);  
      System.out.println("tab1+tab2 ");UtilTab.affiche(tab3);}  
}
```

POO

6

Tableaux à deux dimensions

`int [][] tab;` la référence `tab` qui pointe sur `null`
`tab=new int [2][3];` un tableau de taille 2 de
tableaux d'entiers de taille 3 initialisés par défaut à 0

`tableau[1][1]=5;`

`tableau[0]=new int[2];`

```
import java.io.*;
class Matrice
{   int[][] tab;
    int nb_ligne;
    int nb_col;
    public Matrice()
    {   int i,j;
        nb_ligne=Saisie.lire_entier();
        nb_col=Saisie.lire_entier();
        tab=new int[nb_ligne][nb_col];
        for (i=0;i<nb_ligne;i++)
        {   for(j=0;j<nb_col;j++) tab[i][j]=Saisie.lire_entier();   }
    }

    public void affiche()
    {int i,j;
        for (i=0;i<nb_ligne;i++)
        {for (j=0;j<nb_col;j++)   System.out.print(tab[i][j]+" ");
        System.out.println(" ");   }
    }
}
```

```

class TableauB
{public static void main(String[] argv)
  {boolean[] tableau={true,false,true};
  System.out.println("Deuxieme element de tableau : "+ tableau[1]); }
}

```

/*A l'execution, on obtient :
 Deuxieme element de tableau : false
 */

Tableaux d'objets

```

class Point
{private int x,y;

  public Point()
  {x=Saisie.lire_entier();  y=Saisie.lire_entier();}

  public int getx() {return x;}

  public int gety() {return y;}

  public void affiche()
  {System.out.println("x = "+x+" y = "+y);}
}

class Gestion_tab
{ private Point[] tab;
  private int nb_pts_positifs;

  public Gestion_tab( int n) {tab=new Point[n];}

  public void ajout_pts()
  {for (int i=0; i <tab.length;i++) tab[i]=new Point();}
}

```

```

public void calcul_pts_positifs()
{int nb=0, i;
for (i=0;i < tab.length;i++)
    if ((tab[i].getx() >0) &&(tab[i].gety()>0)) nb++;
    nb_pts_positifs=nb;
}

public void affiche()
{for (int i=0; i <tab.length;i++) tab[i].affiche();}
}

public class TestGestion
{ public static void main (String[] args)
  {Gestion_tab t=new Gestion_tab(3);
   t.ajout_pts(); t.affiche();
   t.calcul_pts_positifs();}
}

```

Classe String

- Définie dans le paquetage java.lang
- Méthodes

- public int length();

- public int compareTo(String s);

>0 si la chaîne est après s (ordre lexicographique)

<0 si elle est avant s

=0 si égale

int l= "bonjour ".length();

String s= "abc" ; String s1= "dbc" ;

if (s.compareTo(s1) <0)

- opérateur concaténation: +

Classe StringBuffer

Les chaînes modifiables

- charAt(int i) retourne le ième char
(classe String a aussi cette méthode)
- setCharAt(int i, char c) remplace le ième char par c